

DUAL FINITE STATE MACHINE

نموذج الـ DUAL FSM نظرة عن قرب

بسم الله والحمد لله والصلاة والسلام علي رسول الله وآله وصحبه وإخوانه أجمعين

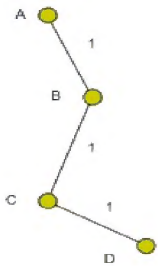
استمراراً في الحديث عن الإصلاحات التي أدخلها جيل الإصلاحيين علي عائلة بروتوكولات الـ Distance Vector، نأتي إلي أهم هذه الإصلاحات، وإن شئت فقل قلب هذه الإصلاحات.

وهذه الإصلاح ينفرد به الـ EIGRP في هذه العائلة العريقة، ألا وهو الـ DUAL Algorithm.

اعتمدت عائلة الـ Distance Vector في حساب الـ Cost علي نموذج "Algorithm"

¹ Bellman–Ford algorithm وهذا النموذج كان هو المسيطر علي عائلة بروتوكولات الـ Routing لفترة طويلة في وقت كانت فيه الشبكات صغيرة، وكان حجم البيانات التي تمر من خلال الشبكة "Traffic" بسيط، وبالتالي لم يكن عامل الوقت "Convergence Time" يمثل مشكلة كبيرة في ذلك الوقت.

ولكن مع توسع الشبكات وزيادة حجمها بدأت تظهر عدة مشاكل واجهها هذا النموذج، علي سبيل المثال:



B	C	D
1	2	3

- مشكلة الـ counting to infinity، فمثلاً هذا الشكل، إذا حدثت مشكلة في الـ link بين A و B، هنا يقوم B بسؤال الراوترات بجانبه عن طريق إلي A، فيجيب C أنه يعرف طريقاً إلي A، ولكن عن طريق

2 hops، فيحاول B الإرسال عن طريق C الذي هو نفسه يعرف الطريق إلي A عن طريق B، وهكذا إلي مالا نهاية.

- مشكلة أخرى هي مشكلة الـ Long Convergence Time، فالوقت الطويل الذي يأخذه هذا النموذج ليتعرف كل راوتر علي الطريق المتاحة إلي شبكة معينة يؤدي إلي حدوث Routing Loop في الشبكة.

- مشكلة الـ Big Overhead نتيجة لزيادة حجم الـ Traffic في الشبكة الذي هو بدوره نتيجة لطريقة عمل الـ Bellman–Ford algorithm، الذي يقوم بعمل broadcast للـ Routing Table كل 30 ثانية، أضف إلي ذلك كبر حجم الـ Routing Table نفسه.

¹ نسبة إلي Richard Bellman and Lester Ford

كل هذه المشاكل وغيرها استدعت، البحث عن طرق لمعالجة هذه المشاكل، من هنا بدأنا نسمع عن الـ Split Horizon mechanism، والـ Split Horizon Reverse، والـ Route Poisoning، وغيرها. ولكن المشكلة أن نظام بسيط كهذا بل وغير متناسب مع الزيادة الكبيرة والمطرودة في حجم أي شبكة قادنا في النهاية وفي بداية التسعينات إلى الـ Dual Update Algorithm².

وقد تم تلافي المشاكل والعيوب المتعددة لنموذج Bellman-Ford algorithm، في هذا النموذج الجديد، بل وأثبت هذا النموذج مع بروتوكول الـ EIGRP أنه الأفضل بلا منازع في عائلة الـ Distance Vector.

وقد تحدثنا في مقال سابق عن طريقة عمل هذا النموذج والمعادلات الرياضية المستخدمة في حساب الـ Cost في هذا النموذج³.

حديثنا اليوم إن شاء الله عن الكوارث!!!

نعم، ماذا يفعل الـ EIGRP إذا حدثت كارثة في الشبكة، مثلاً لا يستطيع الوصول لشبكة معينة، كيف يفاضل بين الـ Routes المختلفة الموجودة عنده، وكيف يختار الأفضل في النهاية.

قبل أن نشرع في هذا المبحث، هناك عدة مفاهيم يجب أن تكون واضحة في ذهن القارئ قبل أن نبدأ.

- Feasible distance (FD): هي أقل metric إلى جهة معينة.

- Reported distance (RD): هي الـ metric التي يستطيع neighbor router الوصول إلى جهة معينة.

- Feasibility condition (FC): وهذا الشرط يتحقق حينما يكون الـ (RD) أقل من الـ (FD).

- EIGRP successor (ES): هو الـ neighbor router الذي تحقق فيه الـ (FC)، وهو المستخدم في الوصول إلى جهة معينة حالياً.

- Feasible successor (FS): هو الراوتر المرشح للاستخدام للوصول إلى جهة معينة في حال أن الـ EIGRP successor، لا يمكن الوصول إليه.

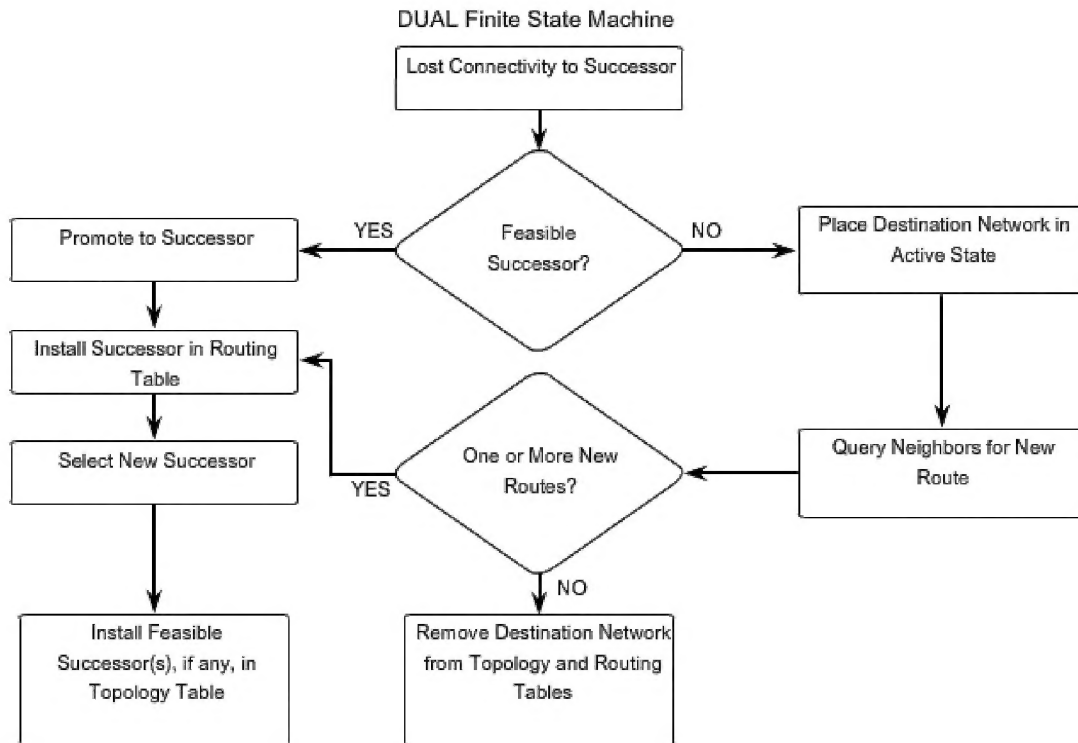
² تم تطوير هذا النموذج في بداية التسعينات علي يد الدكتور J.J. Garcia-Luna-Aceves
³ راجع مجلة الشبكات عدد نوفمبر 2010، مقال بعنوان Eigrp Cost Calculations، أحمد مصطفى.

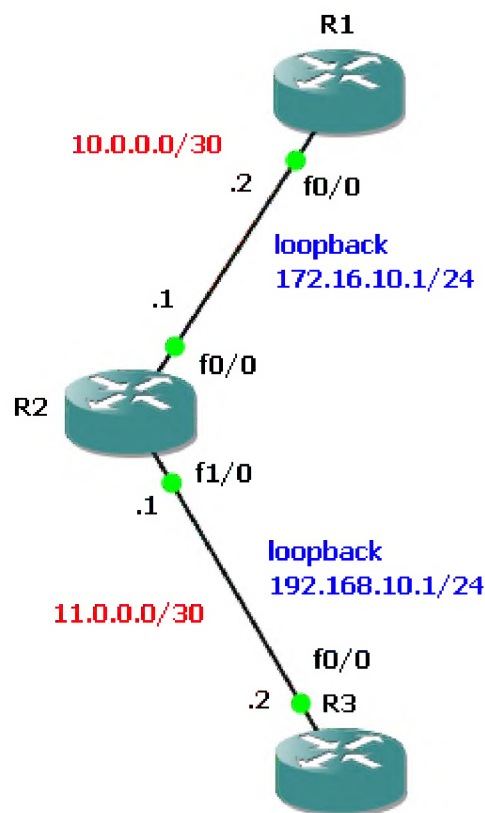
- Finite State Machine (FSM):

هذا المفهوم هو مفهوم عام، وله تطبيقات عديدة في مجالات عدة، فهو يوضح الاحتمالات التي من الممكن أن يمر بها شيء معين، و الأسباب التي تؤدي إلي ذلك، وأيضا النتائج المترتبة علي هذه الأسباب.

ويستخدم هذا المفهوم مع الـ DUAL Algorithm ليوضح كيف يتصرف الـ EIGRP في حالة حدوث Fail للـ (ES).

ويمكن تلخيص الحالات التي يمر بها الـ DUAL Algorithm من خلال نموذج الـ (FSM) من خلال الرسم التوضيحي التالي:





ولنأخذ مثال لتوضيح كيفية عمل FSM إن شاء الله:

إذا كان لدينا هذا الـ Topology، والمفعل عليه بروتوكول الـ EIGRP:

وبالنظر إلى الـ Routing Table في R3، نجد أن يستطيع الوصول إلى شبكة 172.16.2.0/24 عن طريق R2، وهذه المعلومة عرفها عن طريقة بروتوكول الـ EIGRP:

هنا نقوم بتفعيل الـ FSM Debug لنري ماذا يحدث داخل الـ FSM أثناء حدوث تغيير في الشبكة:

```
R3#debug eigrp fsm
```

EIGRP FSM Events/Actions debugging is on

وبفرض أن الـ LINK بين R2، R3 حدث له Fail نجد أن الـ FSM علي R3 بدأ في العمل كالتالي:

```
*Mar 1 00:11:37.339: DUAL: rcvquery: 192.168.10.0/24 via 10.0.0.1 metric 4294967295/4294967295, RD is 158720
```

```
*Mar 1 00:11:37.343: DUAL: Find FS for dest 192.168.10.0/24. FD is 158720, RD is 158720
```

```
*Mar 1 00:11:37.347: DUAL: 10.0.0.1 metric 4294967295/4294967295 not found Dmin is 4294967295
```

```
*Mar 1 00:11:37.351: DUAL: Peer total/stub 1/0 template/full-stub 1/0
```

```
*Mar 1 00:11:37.355: DUAL: Dest 192.168.10.0/24 (Split Horizon) not entering active state.
```

```
*Mar 1 00:11:37.355: DUAL: send REPLY(r1/n1) about 192.168.10.0/24 to 10.0.0.1
```

```
*Mar 1 00:11:38.135: DUAL: Removing dest 192.168.10.0/24, nexthop 10.0.0.1
```

```
*Mar 1 00:11:38.139: DUAL: No routes. Flushing dest 192.168.10.0/24
```

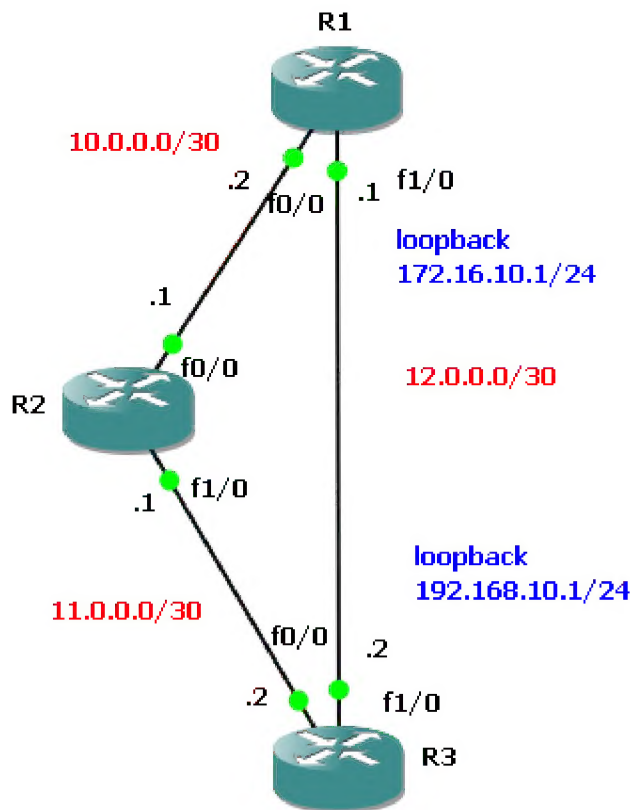
هنا وفي هذه الحالة لما لم يكن هناك أي طريق آخر بالنسبة لـ R1 للوصول إلى شبكة 192.168.10.0/24 قام R1 بحذف هذه الشبكة من الـ Routing table والـ⁴ topology table

وهذا واضح من السطرين الأخيرين في الـ OutPut.

وهذا يوضح الجزء الأول من نموذج الـ FSM، الذي ذكرناه أعلي.

أما إذا كان عندنا أكثر من Route إلى شبكة 192.168.10.0/24 كما في الشكل التالي، تعالوا لنري كيف يتعامل معها الـ DUAL:

أولاً لنري ما هي الـ Routes الممكنة بالنسبة لـ R1 للوصول إلى شبكة 192.168.10.0/24:



R1#show ip eigrp topology all-links

Output Omitted

P 12.0.0.0/8, 1 successors, FD is 28160, serno 6

via Summary (28160/0), Null0

P 12.0.0.0/30, 1 successors, FD is 28160, serno 4

via Connected, FastEthernet1/0

P 192.168.10.0/24, 1 successors, FD is 156160, serno 7

via 12.0.0.2 (156160/128256), FastEthernet1/0

via 10.0.0.1 (158720/156160), FastEthernet0/0

هنا نلاحظ أن R1 لديه طريقين للوصول إلى شبكة 192.168.10.0/24، كما هو موضح في الـ OutPut ولكن المستخدم حالياً للوصول

إلى شبكة 192.168.10.0/24 هو 12.0.0.2 (156160/128256), FastEthernet1/0

عن طريق R3، وبفرض أن الـ LINK بين R1 و R3 حدث له Fail، نلاحظ أن الـ DUAL بدأ بالعمل طبقاً للـ FSM كالتالي:

⁴ وهي الجداول التي يعتمد عليها الـ EIGRP في عمل العلاقات Relationships مع الراوترات المجاورة.

*Mar 1 00:19:26.115: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 12.0.0.2 (FastEthernet1/0) is down: holding time expired

*Mar 1 00:19:26.119: DUAL: linkdown: start - 12.0.0.2 via FastEthernet1/0

Output Omitted

*Mar 1 00:19:27.083: DUAL: Removing dest 192.168.10.0/24, nexthop 12.0.0.2

*Mar 1 00:19:27.091: DUAL: RT installed 192.168.10.0/24 via 10.0.0.1

*Mar 1 00:19:27.091: DUAL: Send update about 192.168.10.0/24. Reason: metric chg

*Mar 1 00:19:27.095: DUAL: Send update about 192.168.10.0/24. Reason: new if

هنا نلاحظ أن R1 حذف الـ Route إلى شبكة 192.168.10.0/24 عن طريق R3 واستخدم بدلاً منه 10.0.0.1 والذي هو R2، ثم بدأ بإرسال التحديثات لهذه الشبكة "192.168.10.0/24" نتيجة لتغير الـ Metric وتغير الـ Next Hop.

وهنا نقطة هامة هل إذا عاد f1/0 علي R3 هل يقوم الـ DUAL بالعمل من جديد؟

الحقيقة نعم، كما يتضح من التالي:

R1#show ip eigrp to

Output Omitted

P 12.0.0.0/8, 1 successors, FD is 28160

via Summary (28160/0), Null0

P 12.0.0.0/30, 1 successors, FD is 28160

via Connected, FastEthernet1/0

P 192.168.10.0/24, 1 successors, FD is 156160

via 12.0.0.2 (156160/128256), FastEthernet1/0

وهذا عكس بروتوكول الـ OSPF، فحتي لو عاد الـ DR بعد حدوث أي انقطاع له من الشبكة، لا يتم انتخابه DR مجدداً في نفس الوقت، بل ينتظر الـ OSPF حدوث انقطاع للـ DR الحالي، ويدخل الـ DR القديم في عملية الـ Elections من جديد، وهذه نقطة هامة للـ OSPF لأنه يستخدم في شبكات أكبر وأوسع من الشبكات التي يستخدم فيها الـ EIGRP.

هذا والله أعلي وأعلم

وصلني الله وسلم وبارك علي المصطفى وآله وصحبه وإخوانه أجمعين

محرم، 1432

وكتبه أحمد مصطفى